

# **Object Oriented Programming In Visual Basic® .NET**

**Alan Carter  
Lead Program Manager  
Visual Basic  
Microsoft Corporation**

**5-326**

Microsoft®  
**PDC** 2000  
Professional Developers Conference

Microsoft®  
**.net**

the defining

***point***

# **VB.NET Language**

## **Goals**

- **Modernize**
- **Simplify**
- **Reduce programming errors**
- **Full access to .NET Framework**

# Inheritance

- **Reuse by composition**
- **Reuse by inheritance**
- **Inheritance features:**
  - **Single base class**
  - **Method and property overriding**
  - **Protected access type**

# Inheritance And Versioning

- Inheriting classes easily broken when base class changes
- By default methods are not overridable
- Overriding is explicit using “Overrides” keyword
- “Shadows” keyword used to hide inherited members
- Classes can be declared as NotInheritable



# Encapsulation

- **Shared members on classes**
  - Shared between all instances of a class
  - Previously were defined outside class in a standard module
- **Nested Type Support**
  - Classes and other types can be encapsulated within a class
- **Structures**
  - Support Properties, Methods, and Events

# Structured Error Handling

- Eliminates need for GoTo
- Supports typed exceptions
- Try Construct
  - Guarded code between Try and first Catch statement
- Catch Statements
  - filters on type or general condition
- Finally block
  - always executed
- Throw statement
  - Generates typed exception

The background is a solid green color. On the left side, there are three large, semi-transparent green circles. Thin, light-green lines curve around these circles, creating a sense of motion or a path. The word "demo" is positioned on the right side of the slide.

# *demo*

## **Inheritance**



# Simple And Consistent

- **Eliminate parameterless default properties**
- **Only one form of assignment**
- **Always use parentheses around args**
- **Fix variable declaration syntax:**  
`Dim x, y As Integer`
- **New operator same for all classes**
- **Short-circuiting And and Or operators**

# Writing Robust Code

- **Parameterized constructors**
- **Variables can be initialized where declared**
- **Circular References**
  - **Garbage collector recovers instances in reference cycles**
- **Decimal type replaces Currency**

# Robust Type Checking

- **Option for stricter type checking**
  - **Disallow Implicit conversions that may fail at runtime**
  - **Disallow late-bound calls**
- **Overloaded methods supported**
- **No unexpected implicit conversions**
  - **Strings and numeric**
  - **Date and Boolean**
- **Array declarations must specify rank**

# Full Support For .NET Frameworks

- **First-class access to framework**
- **Seamless Language Interoperability**
- **File I/O names not reserved**
- **Importing Namespaces**
  - **Allows unqualified access to types**
  - **Placed at file or project level**

# Raw Power

- **Delegates**
  - Object oriented function pointer
  - Points to a specific method of a specific instance
- **Event Extensions**
  - Implemented on top of delegates
  - Dynamic hook-up of method as handler
  - Many events can be routed to same method
- **Free Threading**
  - SyncLock statement for

The background is a solid green color. On the left side, there are three large, semi-transparent green circles. Thin, light-green lines curve around these circles, creating a sense of motion or a path. The word "demo" is positioned on the right side of the image.

# *demo*

**Free Threading**



# Performance

- **Reference assignment for arrays**
- **Instance size much smaller**
- **Faster Object manipulation**
  - **Faster creation**
  - **Faster assignment**
  - **Supports direct data member access**

The background is a solid green color. On the left side, there are three large, semi-transparent green circles. Thin, light-green curved lines connect these circles, creating a network-like pattern that extends across the left half of the image.

# *demo*

## **Performance**

# Summary

- **The features you asked for!**
- **Modern, Simple, Consistent, and Robust**
- **Full access to the .NET platform**
- **Your skills and investments will move forward**

# **Related Sessions And References**

- **Other PDC sessions recommended:**
  - **5-322 What's New In Visual Basic .NET**
  - **5-313 Introduction to C#**
- **References**
  - **Visual Basic Language Reference**
  - **Visual Basic Language Specification**

# Discussion



Where do **you** want to go today?

**Microsoft**



Microsoft®  
**PDC** 2000  
Professional Developers Conference

Microsoft®  
**.net**™

the defining

***point***